Digital Government Academy Course:
Enterprise Content Management

Presented by Interwoven, Inc, for the State of Washington

Today's Agenda

- Today we'll continue our exploration of Enterprise Content Management in the State of Washington
- Today's topics:
  - **TeamSite Security**
  - Workflow Analysis
  - Workflow Design

- **Objectives:**
  - TeamSite authentication security
  - TeamSite roles management options
  - Branch security issues
  - File permission issues
- **Action planning:**
  - Authentication model planning
  - Roles management
  - Cross-agency branch access requirements
  - File permission requirements

Today's Agenda

- Today we'll continue our exploration of Enterprise Content Management in the State of Washington
- Today's topics:
  - TeamSite Security
  - **Workflow Analysis**
  - Workflow Design

- **Objectives:**
  - Workflow requirements analysis
  - Workflow capabilities
  - Workflow design best practices
- **Action planning:**
  - Updating and refining your workflow business process diagrams
  - Identifying inter-agency workflow dependencies

Today's Agenda

- Today we'll continue our exploration of Enterprise Content Management in the State of Washington
- Today's topics:
  - TeamSite Security
  - Workflow Analysis
  - **Workflow Design**

- **Objectives:**
  - Workflow model diagramming
  - Workflow task types
  - Task concurrency
- **Action planning:**
  - Creating detailed workflow diagrams from business process diagrams

Seminar Schedule

- Day 1: December 4, 2002
  - Course Kickoff
  - ECM
  - TeamSite Templating
- Day 2: December 11, 2002
  - Develop-and-Deploy
  - TeamSite in the Enterprise
  - Branching Structures
- **Day 3: December 18, 2002**
  - TeamSite Security
  - Workflow Scenarios
  - Designing a Workflow
- **Day 4: January 8, 2003**
  - Designing Data Capture Forms
  - Designing Presentation Templates
  - Re-use via Templating
- **Day 5: January 15, 2003**
  - TeamSite and Metadata
  - Finding your Assets
  - Supporting Personalization
  - Course Summary

TeamSite Security

Topic Objectives

- In this topic, we'll cover:
  - TeamSite authentication security
  - TeamSite roles management options
  - Branch security issues
  - File permission issues
- The action planning at the end of this topic will be:
  - Authentication model planning
  - Roles management
  - Cross-agency branch access requirements
  - File permission requirements

TeamSite Security Features

- To access TeamSite, a user must have a UNIX user account on the local server, and have a TeamSite **Role**
- TeamSite provides two levels of security
  - Authentication
    - Regulates access to the TeamSite web user interface and some CLTs
    - Controlled by TeamSite roles
  - File and directory access
    - Regulates access to web content
    - Controlled by setting branch, workarea, file and directory permissions
- TeamSite can use normal HTTP protocol or secure HTTPS SSL connections

TeamSite Roles

- The 4 TeamSite roles (Master, Administrator, Editor and Author) control a user's ability to log on and interact with TeamSite
- The default role management system for TeamSite are the **role files** in ***iw-home/conf/roles***
  - These are simple text files and can be modified with a text editor, with CLTs, or via the administrative web interface
- Alternatively, TeamSite can integrate with an LDAP server, and roles can be managed within the directory database
  - Advantage: centralized role management, simplified administration

Roles Management Issues

- The roles files must be kept current with your organizational model
- If you don't use LDAP, you need some method to keep your roles consistent with changes in personnel, jobs, projects, etc
- Options:
  - Manually change/add user roles as they occur
  - Use a timed script to revise the roles files based on group membership
    - For instance: all members of the webdev team automatically made Editors, all members of the sysadmin group automatically made Masters, etc
- Be careful of licensing issues when modifying the roles files

TeamSite SSL Support

- Users can connect to TeamSite via SSL
  - Use **`https://servername/iw`** to connect securely
- TeamSite ships with a default set of SSL certificates
- If desired, you can create new certificates
  - Use an Internet certificate authority, such as www.verisign.com
  - Copy the certificate files into ***`iw-home`*`\iw-webd\conf\ssl.cert`** and **`ssl.key`**
  - Edit ***`iw-home`*`\iw-webd\conf\httpd.conf.template`** to list the new certificate files
  - Run **`iwreset -ui`**

Controlling Branch Access

- Once a user logs on to TeamSite, s/he will have access to any allowed branch
  - However, they can only change content in a branch that contains a workarea they can access
- Normally, once a TeamSite user logs on they will see all branches and workareas—even those they have no access to
- To control access to branches, set **read permission** on the corresponding branch directory in the IFS to only the group that should have access to it
  - for instance, `/iwmnt/default/main/`**`safari`**
  - On Windows, you can any combination of user and group permissions to a branch

Branch and Workarea Security

- In general, a user should only see the branches s/he would reasonably need to access
  - This will make it easier for them to find their own branches
- **Branch Security**
  - If the **iw.cfg** file has **branch_security=on** in the **[iwserver]** section, then users can only see the branches for which they have read access
- **Workarea Security**
  - If the **iw.cfg** file has **workarea_security=on** in the **[iwserver]** section, then users can only see the workareas for which they have read access, either via ownership or membership in the workarea's group for sharing

Controlling File Access

- Once logged in and with a branch selected, a user's rights on files in the TeamSite file system are based on the files' permission settings
- These permissions are set in the same way as for the host operating system
- You may need to further restrict users' access to files based on the user's function in the organization
- On Windows this means using the **CACLS** command or Windows GUI tools to set file permissions
- This must be done inside a workarea, and then submitted

Example: File Permissions

- In a particular organization the following file and directory permission rules might apply:
  - Only members of the **webdev** group should be able to modify **.js** files
  - Only members of the **marketing** team can change the contents of files in the **/corporate/...** directory
  - In the intranet branch, the contents of the **/hr/payroll/...** directory must be hidden from all employees except those in the **hr** team
- We can use file-level permissions to enforce all the above rules

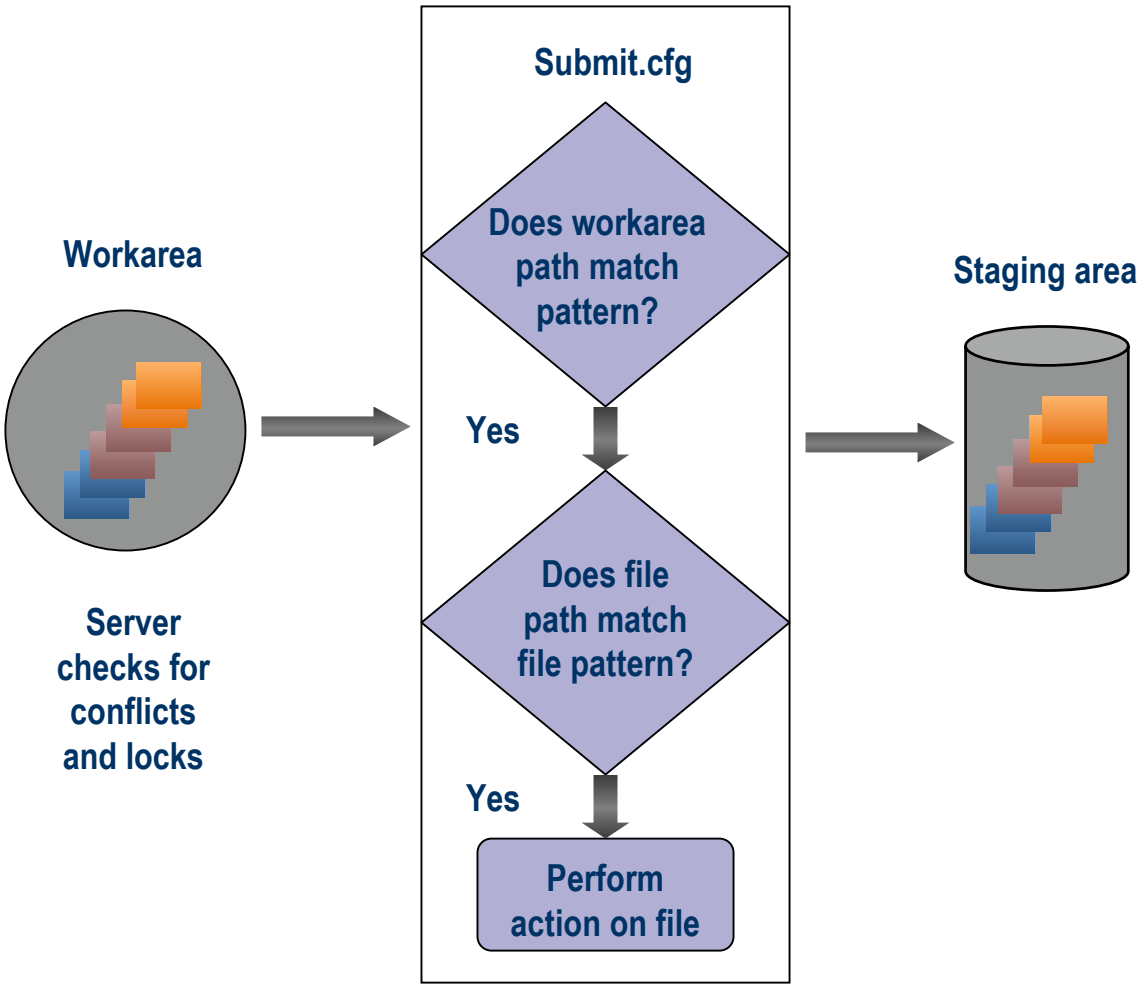
TeamSite File Permissions

- TeamSite uses normal Windows access permission settings for files and directories
- To restrict access to files or directories, set their permissions to allow access only by those users and groups that should have access
- Remember that a directory must have read and execute permission to be viewable and navigable by a user
- On Windows, be careful of permission inheritance
  - By default, the IFS mimics a Windows NTFS file system, which defaults to inherited permissions
  - Thus any changes to your workarea root directory will be duplicated on all sub-directories and files
  - You may want to disable this

Submit Filters

- Use submit filters to automatically change file attributes, such as a file or directory's owner, group, and permissions immediately before files are submitted to the staging area
- Submit filters can enforce your file security and access policies during development
- Submit filters are defined in ***iw-home*/local/config/submit.cfg**

How a Submit Filter Works



Deployment Security

- When content moves from TeamSite to a production server, it often does so via the Internet
- OpenDeploy can be configured to use any of several increasingly-secure modes of transmission
- The highest level, SSL 128 bit encryption, insures your data is as secure as possible as it moves from one point to another

File Permissions: Development v. Production

- Typically you want to set permissions according to one set of policies during development, and a different set of policies during production
- Example: **.html** files should be FULL ACCESS for the **webdev** team during development, read-only for everyone else, but read-only for *everyone* on the production server
- Use **submit.cfg** to establish development permission policies
- Use **OpenDeploy permission filtering** to establish production permission policies

Washington State Cross-Agency Access

- You must consider cross-agency access between branches when establishing your security model
  - Which agencies collaborate?
  - Should they access each others' branches?
  - Or should there be a common branch for shared content?

Internets/Intranets/Firewalls

- If the TeamSite server will be accessed through a firewall, some type of arrangement must be made to permit this
  - Users connecting from home or from other office
  - Upload/download of files via LaunchPad
  - OpenDeploy deployments
- Two common approaches:
  - **Router-based**: open "pinholes" (ports) on the corporate firewall to allow access to the TeamSite and OpenDeploy servers—inexpensive, but can be hard to administrate
  - **Virtual Private Network**: Use VPN servers and client software to control access—more expensive, but very flexible

Action Exercise: TeamSite Security Requirements

- Within your group, spend 20 minutes discussing the following:
  - How will we keep the roles up-to-date?
  - What kind of cross-agency branch access do we expect?
  - What file permission policies do we want to enforce
- After discussion, spend 20 minutes documenting the following:
  - Your authentication plan
  - Cross-branch access requirements
  - File permission policies (as many as time permits)
- Each group will then present their findings to the class one at a time

Action Item Discussion

- Class presentation
- Questions
- Take a few minutes to consider other group action items
  - Integrate theirs with yours if needed

Analyzing Workflow Scenarios

Topic Objectives

- In this topic, we'll cover:
  - Workflow requirements analysis
  - Workflow capabilities
  - Workflow design best practices
- The action planning at the end of this topic will be:
  - Updating and refining your workflow business process diagrams
  - Identifying inter-agency workflow dependencies

TeamSite Workflow

- On the first day of this seminar series, we discussed workflow briefly
- Today we'll return to that topic and cover it in more detail
- You may want to review the business process diagrams you made on the first day

Review: What is a Workflow?

- A **workflow** is a model of a business process
- A workflow is a **template**, which can be used to start workflow **jobs** for specific activities
  - One workflow template can launch any number of jobs, either one after another or at the same time
- A workflow starts in response to a business event, and carries out some kind of business-relevant operation
  - The initiating event might be: new press release, change to product or service, or website update ready to go live
  - The business operation might be the publication of the new press release or product announcement, or the deployment of a new release of the website

What can TeamSite Workflow Do?

- A TeamSite workflow can:
  - Carry out automated tasks without user involvement
  - Notify users that they have a task to do, and give them any needed instructions
  - Perform notification via email or any other means, as long as a computer interface to the notification method exists
    - Examples of other notification methods: pager, portal page, instant messenger service, etc
  - Give each part of the workflow a time limit, and carry out optional tasks if the time limit expires
  - Perform repeated or time-initiated tasks
  - Execute tasks serially or in parallel

Examples: Workflow Scenarios

- 1. Contributors update product information on the web site
  - Create three new pages
  - Modify ten pages
  - Many interrelated links
  - Routes entire collection for review and deployment
- 2. New product section needed for web site
  - Manager initiates workflow
    - Task #1: Create new HTML pages for new product section
    - Task #2: Create new images for product pages
  - Web publisher generates HTML content as web designer generates images
  - Indicate which tasks are completed and routed for review

How Do Users Interact With Workflow?

- Typically, users participate in a workflow by either editing/creating files or by reviewing/approving the work of others
- The following options exist for workflow-user interaction:
  - The tasks tab in the WebDesk author UI
  - The To-do display in the WebDesk Pro UI
  - CCI workflow links integrated into a portal web page
  - HTML-based email messages with embedded workflow CCI links
- Of these, the email interface is perhaps the most useful for casual users, such as reviewer/approvers, etc
  - It incorporates both notification and user interface in one package

Examples: Workflow Scenarios (continued)

- 3. Major project release to Internet web site
  - Routed through project leaders
  - Routed through respective department heads (legal, marketing)
  - Routed through company CEO for final approval and deployment
- 4. Daily content update to web site
  - Press releases reviewed by legal, marketing
  - Product information reviewed by marketing and engineering
  - CEO can review and override any approval process
  - Each reviewer emailed reminder if past due

Common Workflow Uses

- Submit
  - There are default submit workflows built into TeamSite
- Assign
  - To be used by a workarea owner (editor) to assign work to a participating author
  - Includes email notification, review by workarea owner, and submit
- Templating
  - To automate the review of new content and generation of output pages
- Deployment
  - To send completed content to a production server

Workflow Pre-Design Issues

- Who will use the workflow?
- Will the workflow apply to one branch or several?
  - In the State of Washington government, will some agencies use the same workflows?
- Who will typically start a workflow?
- What role will reviewer/approvers work in?
  - Will they be able to make small changes, or only review and comment?
- Who should own a workflow job?
  - The creator of the job, or a standardized "job administrator"?

Issues to Consider in the State Government

- Which workflows are unique to your agency?
- Which workflows are shared?
- Would any of your workflows cause a workflow in another agency to start executing?
  - Or vice versa?
- What notification methods are available?
- Who should be notified if a job is not completed on time?

Review/Approval

- Nearly all workflows involve review or approval
- What are the consequences of a review?
  - Approve
  - Reject
  - Cancel
  - ???
- Who will be doing reviews?
- Are **all** reviews required, or are some optional?
- Can reviews by different reviewers be done in parallel, or must they be done one after another?

Workflow Development Best Practices

- KISS, at first
- Start small: look for the 20% of the requirements that cover 80% of the scenarios
  - Don't get bogged down in the corner-cases
- Get an early version of your workflow done, reviewed and implemented quickly, so you can evaluate it and adjust
  - Don't over-design your first increment, only to have to tear it down and start over later

Workflow Development Sequence

1. Gather requirements
2. Draw a simple business-process diagram
3. Review the diagram with experts and validate
4. Convert the diagram to workflow models (next topic)
5. Create the workflow template using WorkflowBuilder or by authoring a WFT file
6. Implement the workflow template and test
7. Refine and extend as needed over time

Action Exercise: Workflow Scenario Analysis

- Within your group, spend 20 minutes discussing the following:
  - Review the workflow scenarios from day 1
  - Identify any additional workflows you've thought of
  - Consider cross-agency workflow issues—reuse, sharing, workflow interdependencies
- After discussion, spend 20 minutes documenting the following:
  - List the workflows by name, in priority order (ie, which ones do you need first?)
  - As time permits, diagram the workflows, expanding on what you had from day 1 in the light of what we've discussed now
  - List any cross-agency workflow issues you've identified
- Each group will then present their findings to the class one at a time

Action Item Discussion

- Class presentation
- Questions
- Take a few minutes to consider other group action items
  - Integrate theirs with yours if needed

Workflow Design

Topic Objectives

- In this topic, we'll cover:
  - Workflow model diagramming
  - Workflow task types
  - Task concurrency
- The action planning at the end of this topic will be:
  - Creating detailed workflow diagrams from business process diagrams

Workflow Design Process: 1

- 1. Determine the workflow requirements
  - Determine the structure that defines the tasks and their interdependencies
  - Which tasks need to be performed
  - Map tasks to task types
  - Is there an order in which these tasks must be performed or can they be performed concurrently?
- 2. Determine basic task attributes
  - Can the owners of the tasks be mapped to TeamSite roles?
  - Are the tasks read/write or read-only?
  - Does the task always take place in the same workarea?

Workflow Design Process: 2

- 3. Create the TeamSite Workflow design diagram
  - Map out task types and transitions
  - Assign task-names to each task
  - Assign descriptive names to transitions
  - Note non-default attribute settings
- 4. Implement the design
  - Create a Workflow Template File (.wft)
  - Include the template in the GUI (**available_templates.cfg**)
  - Test and debug the workflow

Definitions

Term	Definition
Workflow template	file containing a description of a type of workflow
Job	an active workflow
Task	one unit of work within a workflow
Successor	a task that follows after another task in a workflow
Predecessor	a task that comes before another task in a workflow
Transition	To end one task and start another
Instantiate	To start a new workflow job

Workflow Diagramming

- Before you develop a workflow template you should diagram it as part of the design process
- Visio makes a good diagramming tool if available
- You can download a Visio stencil set for Interwoven workflow from http://devnet.interwoven.com
  - Search for "visio stencil"
- As you refine your workflow during iterative design and testing, make sure you keep your diagrams up-to-date

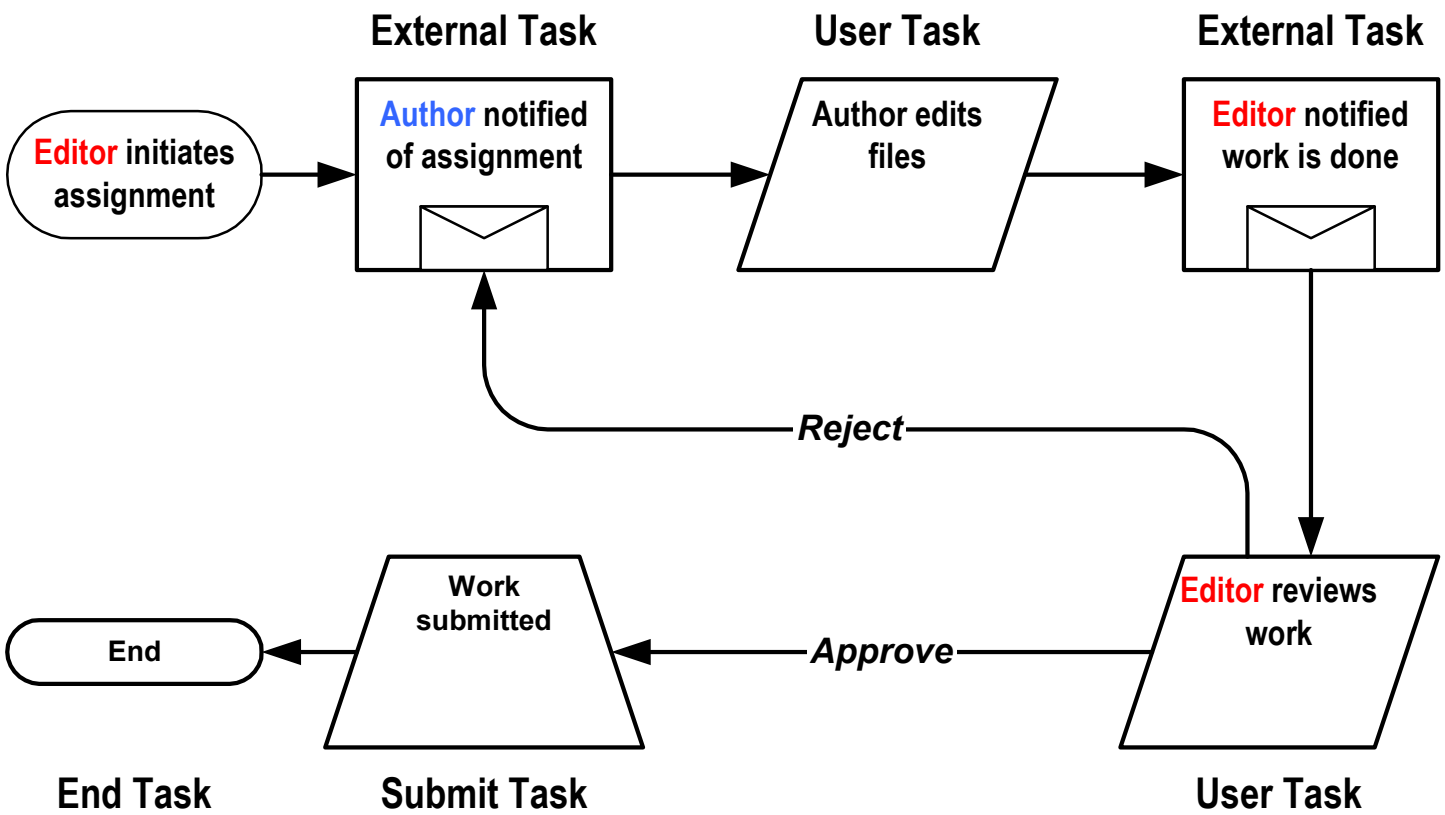
Common Task Types and Uses: 1

Purpose	Task Type
Notify a user or group that they have an assigned task	**emailtask**
Edit, create or review files by a person	**usertask**
Assign work as per a user task to a group of people	**grouptask**
Copy files from one area to another in TeamSite	**updatetask**
Lock files	**locktask**
Unlock files	**externaltask** (calling unlock.ipl)
Perform automated tasks without user interaction (template generation, etc)	**externaltask**

Common Task Types and Uses: 2

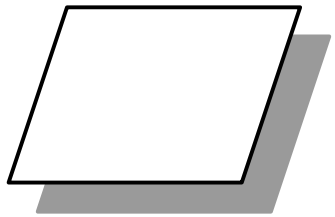
Purpose	Task Type
Perform automated tasks requiring user interaction	**cgitask**
Launch another workflow job	**wftask**
Submit files to the staging area	**submittask**
Publish an edition or deploy	**externaltask**
Writing a job log	**externaltask**
Terminate a workflow	**endtask**

Common Task Diagram Symbols

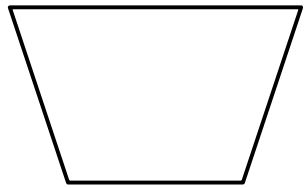


Other Kinds of Task Symbols

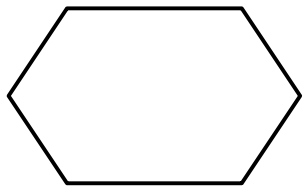
Group Task



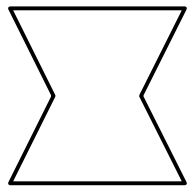
Update Task



CGI Task

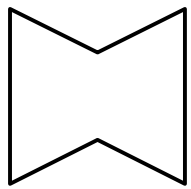


Dummy Task

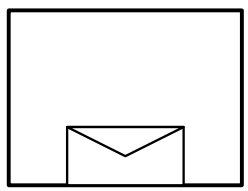


Nested Workflow Task

Lock Task

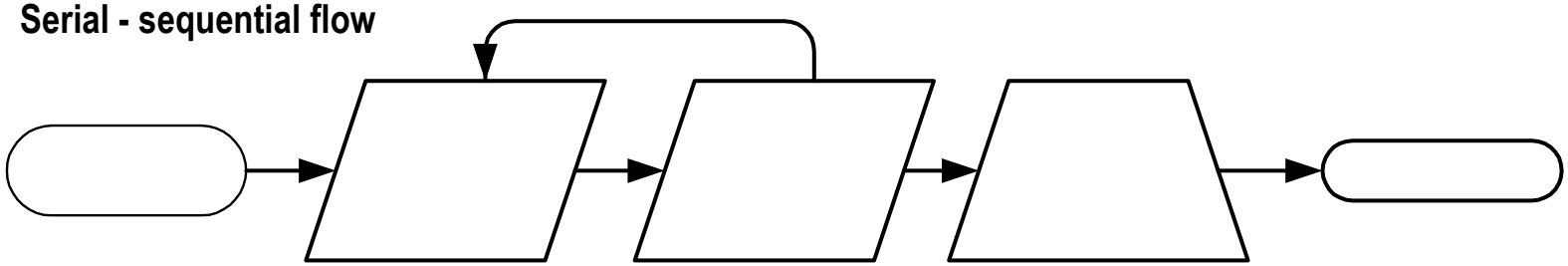


Email Task

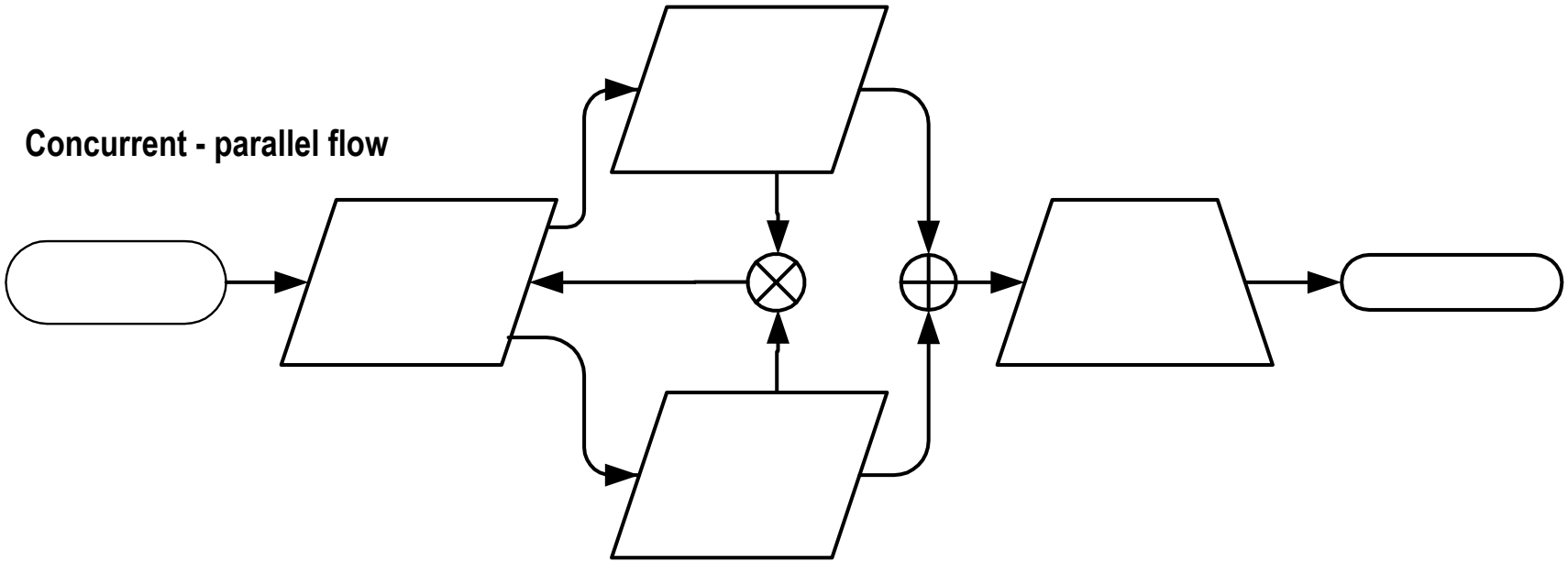


Typical Workflow Scenarios

Serial - sequential flow



Concurrent - parallel flow



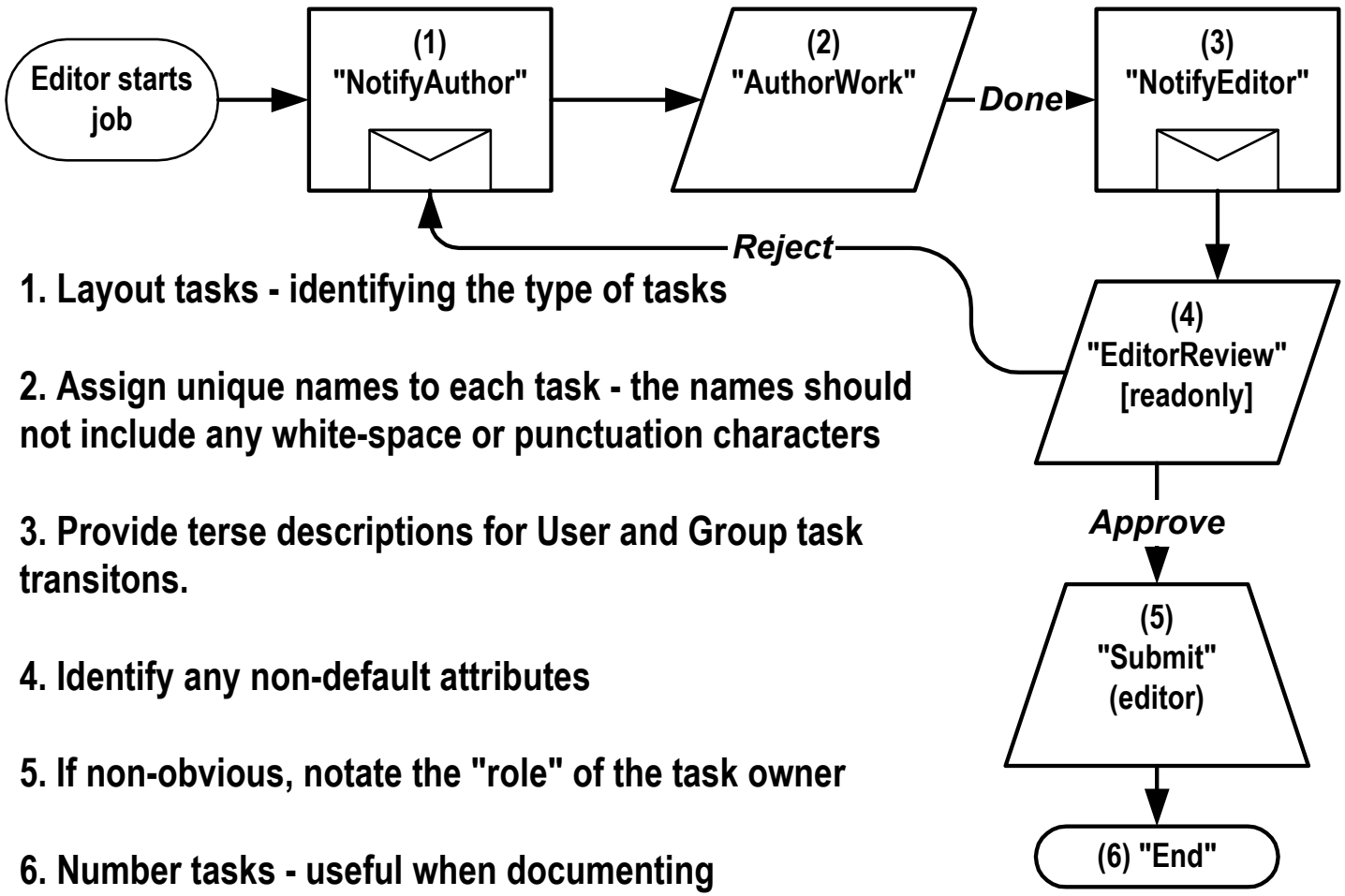
Concurrency in Workflow

- Note that on the previous slide we indicated that more than one task can be active at a time
- This is achieved simply by having more than one successor for a task
  - When the task is completed, all of its successors will activate
- Each transition option for a task can specify one ***or more*** successors
- If there are multiple successors, they will run in parallel
- Potentially useful, job takes less time overall, but might be difficult to plan for

Multiple Predecessors

- Note that when a task has more than one ***predecessor***, the rules for activation are more complex
- If the predecessors are in an **and** relationship, the task will activate only when **all** predecessors are complete
- If the predecessors are in an **or** relationship, the task will activate as soon as **any** of the predecessors are complete

Drawing a Workflow Model



1. Layout tasks - identifying the type of tasks

2. Assign unique names to each task - the names should not include any white-space or punctuation characters

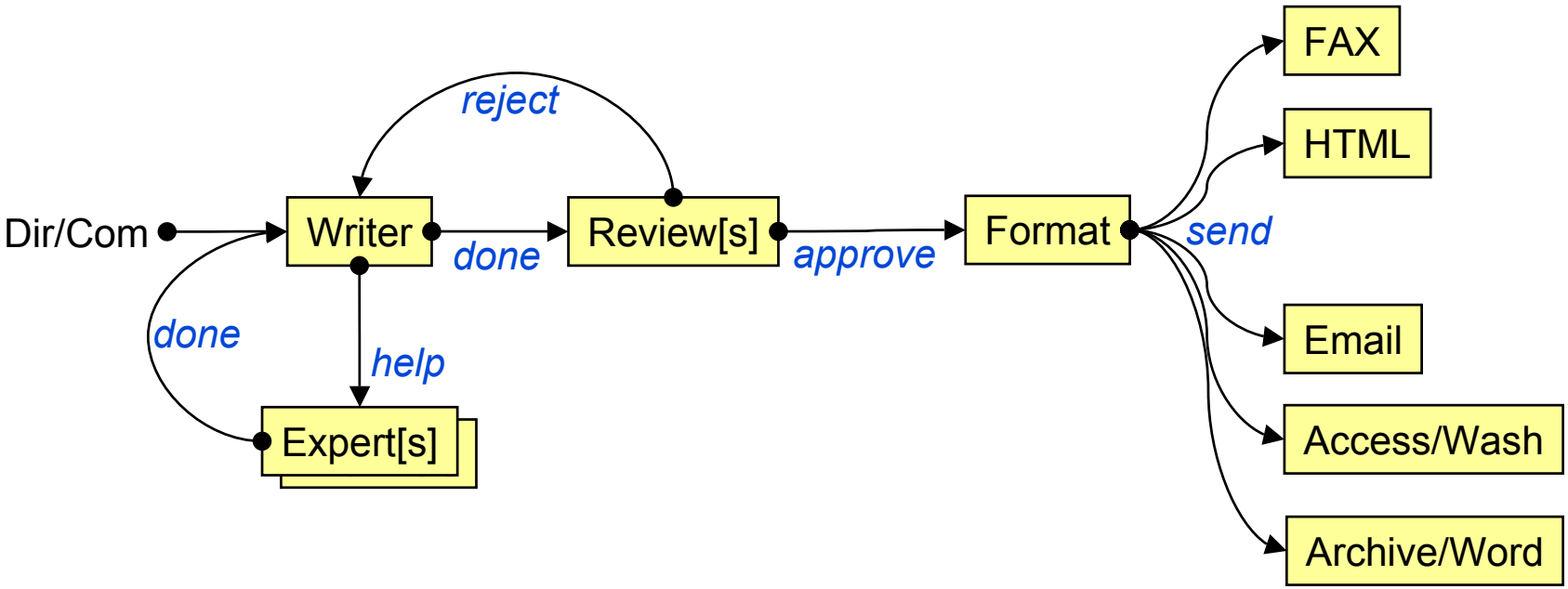
3. Provide terse descriptions for User and Group task transitons.

4. Identify any non-default attributes

5. If non-obvious, notate the "role" of the task owner

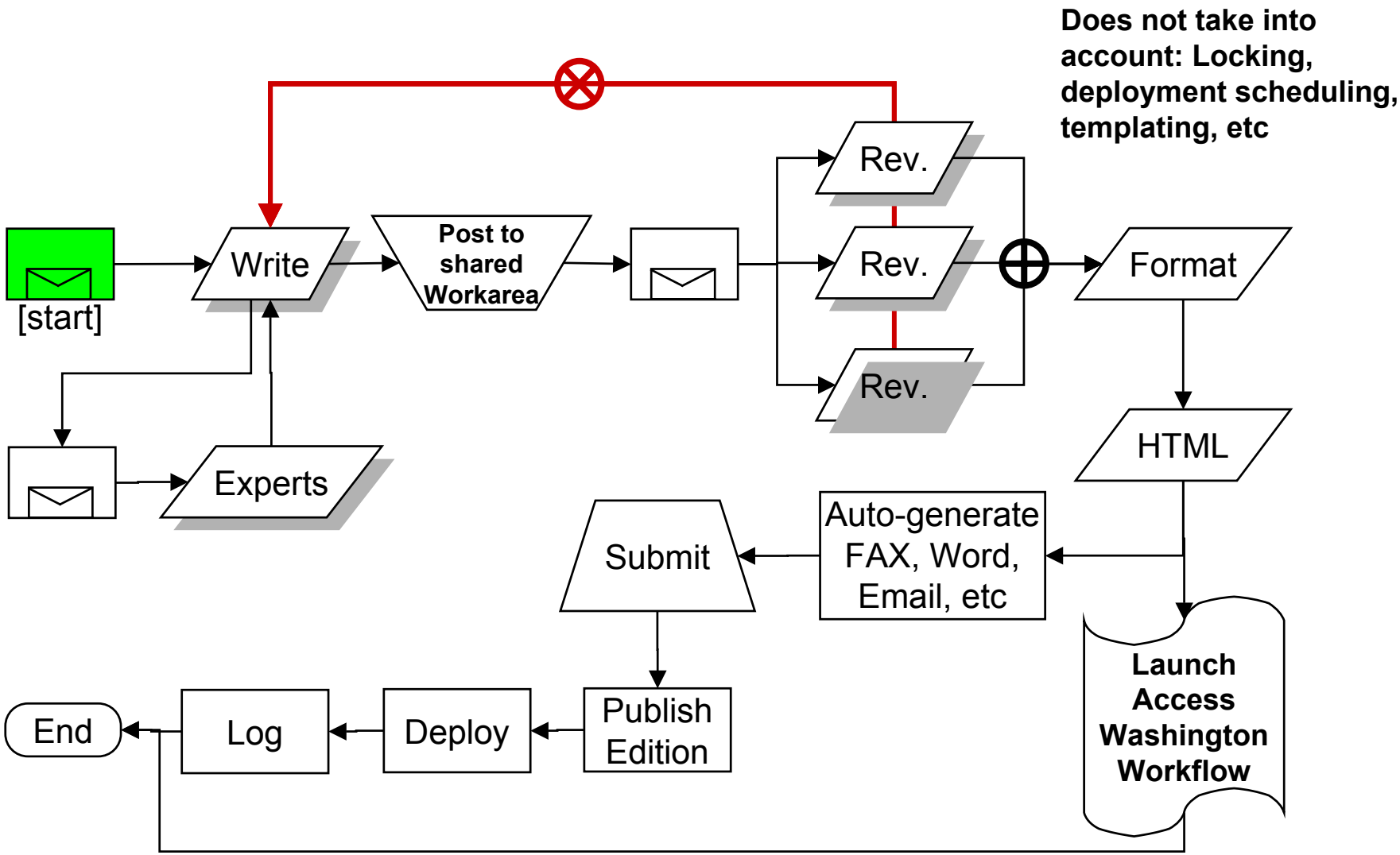
6. Number tasks - useful when documenting

DSHS Example: Press Release--Workflow Model



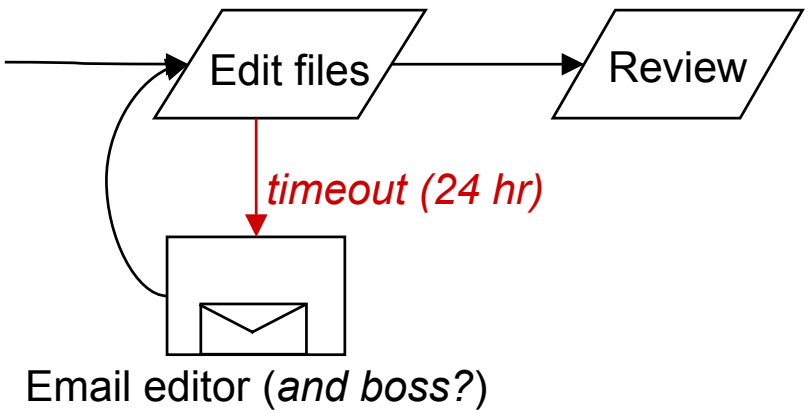
- This is a **business process** workflow model
- Next step: create a **TeamSite workflow diagram**

DSHS Example: Press Release Workflow (*partial*)



Timeouts in Workflows

- Each task can have a **time-out** setting
  - The timeout can be either a time delay in hours and minutes starting from the task's activation, or a set date+time
- If the timeout expires before the task is completed, the task automatically deactivates and the successor[s] specified in the timeout are activated



Workflow Design Guidelines

- Convert your business process diagrams into TeamSite workflow diagrams
- Translate business terms into TeamSite terms
- Take into account technical requirements:
  - Notification tasks
  - Submit
  - Publishing editions
  - Locking and unlocking
  - Timeouts
  - Logging

Action Exercise: Workflow Diagramming

- Within your group, spend 20 minutes discussing the following:
  - What will it take to convert your business process diagrams into workflow model diagrams?
  - What are the high-priority issues, and what can wait?
- After discussion, spend 20 minutes documenting the following:
  - Draw a workflow model diagram in as much detail as possible for at least one of your business process diagrams, taking into account notification, timeouts, and TeamSite technical requirements like submit, publish and deploy
  - Draw as many more as time permits
- Each group will then present their findings to the class one at a time

Action Item Discussion

- Class presentation
- Questions
- Take a few minutes to consider other group action items
  - Integrate theirs with yours if needed

End of Session

- This concludes today's session
- Next session: **January 8, 2003**
  - Designing Data Capture Forms
  - Designing Presentation Templates
  - Re-use via Templating